



How to control Resource allocation on pSeries multi MCM system

Pascal Vezolle
Deep Computing EMEA
ATS-P.S.S.C/ Montpellier FRANCE



Agenda

- AIX Resource Management Tools
 - **WorkLoad Manager (WLM)**
 - **Affinity Services**
 - **Bind command and Resource Set**
 - **Memory Affinity**
- Number of MCMs impact on Parallel job performance
- Versatile System Resource Allocation and Control
 - PSSC tool integrating AIX resource capabilities in HPC environment

Customer Resource control requirements versus AIX capabilities

- **Dedicated physical resources for applications or user groups**
 - Solution 1: Partitioning
 - Solution 2: WorkLoad Manager (WLM) based on Resource Set

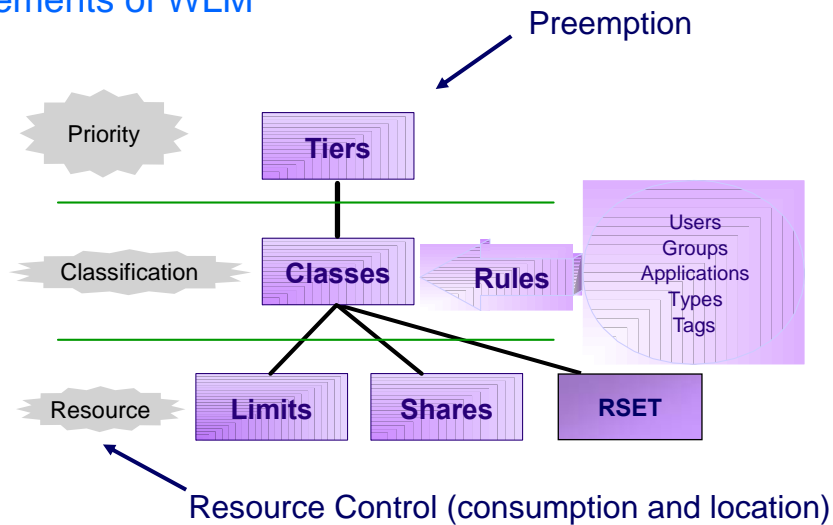
- **Control Resources allocation on standalone system**
 - **Control resources consumption** (CPU, memory, IO)
 - Solution: WLM classes (Consumable features in LoadLeveler)
 - **Control interactive and batch jobs ratio**
 - Solution: WLM & LoadLeveler
 - **Control job priority: Preemption** (i.e. free resources for running high priority jobs)
 - Solution: WLM class tiers&shares or LoadLeveler Gang scheduler (2005 with backfill scheduler)

- **Optimize performances for HPC applications**
 - Memory affinity
 - Binding and Resource Set attachment to guarantee MCM Affinity

Dedicated physical resources

- **What Tools are Available?**
 - **Partitioning**
 - Subdivide a larger machine into several smaller servers
 - Allows multiple OS instances to peacefully coexist
 - Static - AIX 5.1, Dynamic - AIX 5.2
 - Resources includes: CPU, Memory, I/O adapter (Granularity: 1CPU, 256MB memory, 1PCI slot)
 - **drawbacks in HPC: no Memory Affinity (except for Affinity LPAR), no shared IO and network adapters, limited SMP capabilities for multi-threads applications**
 - **AIX Workload Manager**
 - Controls access to the resources of a single AIX instance
 - Allows multiple workloads to peacefully coexist on one AIX image
 - Limits the CPU, Memory and disk I/O bandwidth consumption
 - Granularity: Percentage of CPU time, physical memory, and disk I/O bandwidth

Elements of WLM



WLM shares and limits

- Very useful to control interactive usage while insuring Batch resources (shares) and job priority

- Example on p690 32 processors
 - Requirements (slovakian, tunisian, hungary met):
 - guaranty 8 CPU for interactive and 24 CPU for batch
 - 1 ultra priority batch class for urgent jobs

One WLM solution: 3 WLM classes (default, batchUrgent and batch) with shares and limits

Class Name	batchUrgent	batch	default
Shares	1000	75	25

→ 100%

Hard CPU limit = 75% for batchUrgent class (let 8 CPU free for interactive)

No Urgent job: shares=100, 75 (24 CPUs) for batch and 25 (8 CPUs) for interactive
 Batch Urgent jobs running: shares=1100, 1000 parts for priority job limited to 24 CPUs

AIX Affinity Services

- **Processor Affinity**
 - **2 programming models for CPU binding**
 - 1) Attach a process to a specific CPU ID (with **bindprocessor** command or API ; root and no root user)
 - 2) Attach a process to **Resource Set** (with RSET APIs and commands)
- **Memory Affinity**
 - AIX tries to allocation the memory on the same MCM containing the CPU

1) **bindprocessor process [ProcessorNum] | -q | -u Process**

- **-q** Displays the processors which are available.
- **-u** Unbinds the threads of the specified process.

To bind process pid=999 to processor 1:

```
bindprocessor 999 1
```

To display the processor number where the process is bound :

```
ps -o bnd -p 999
```

Resource set

- **A resource set structure is a set of physical resources:**
 - **CPU** (cpu are identified by a CPU ID created at boot time)
 - Memory pool (current AIX supports only one pool)
- **Attach a process to a rset limits the process to only use the physical resources containing in the rset** (no thread featureb in AIX5.2B)
- **Available with partitioning**
 - System rset 'sys/sys' contains the available CPU and memory pool
 - How to display current rset configuration: **lsrset -a -v**
- **AIX 5.2: Dynamic management capabilities for root and no root users with APIs and AIX commands**
- **2 types of rset: partition rset and effective rset**
 - partition rset: restricted to user root + only one rset per process
 - effective rset: can be used by no root user with **CAP_NUMA_ATTACH** capability:

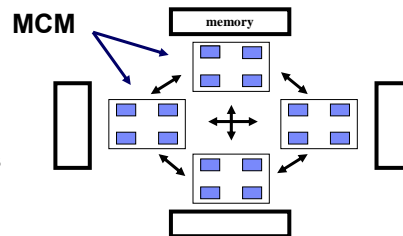
```
> chuser capabilities=CAP_NUMA_ATTACH,CAP_PROPAGATE username
> or add in /etc/security/user file
```

AIX 5.2 Resource Set commands

[http://publib16.boulder.ibm.com/pseries
//publib16.boulder.ibm.com/cgi-bin](http://publib16.boulder.ibm.com/pseries//publib16.boulder.ibm.com/cgi-bin)

- Create a rset:
 - **mkrset** -c CPUlist [-m MEMlist] rsetname
(create a rset for MCM0: **mkrset -c 0-7 test/mcm0**)
- Remove a rset:
 - **rmrset** rsetname
- Display information about rset:
 - **lsrset** [-f] [-v] [-o] [-r rsetname] [-n namespace] [-a]
- Attach (detach) a process to a rset:
 - **attachrset** [-P] [-F] rsetname pid or [-P] [-F] [-c CPUlist] [-m MEMlist] pid
 - **detachrset** [-P] pid
- Execute a command in a rset:
 - **execrset** [-P] [-F] -c CPUlist [-mMEMlist] -e command [parameters]
 - or **execrset** [-P] [-F] rsetname [-e] command [parameters]

Memory Affinity



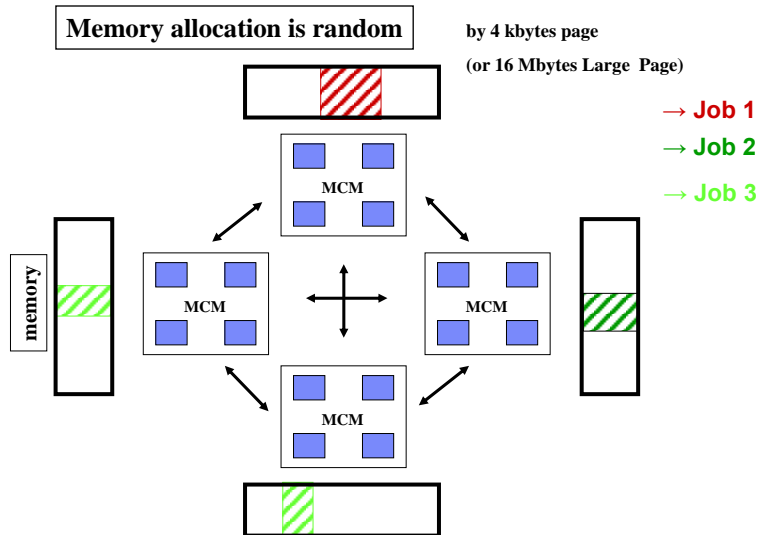
- On pSeries MCM system the memory is attached to the MCMs
- Local memory access is faster
- The target is to improve performance of HPC applications by backing the data in memory that is attached to MCM containing the CPU
- If memory affinity is enabled, AIX managed memory as a set of pools (one per MCM or SCM)
 - Pools can be monitored by the following commands
 - kdb
 - > mempool *
 - > vmpool
 - > free

How to set Memory Affinity

- In AIX 5.1:
 - **vmtune -y 1 or 0 (default)**, (+ bosboot -a, reboot)
 - Global availability for all processes: on or off

- In AIX 5.2:
 - **vmo -memory_affinity=1 or 0 (default)**, (+ bosboot -a, reboot)
 - + variable environment **MEMORY_AFFINTY** provided memory affinity for a selected processed (also available with AIX 5.1G)
 - Two valid settings for MEMORY_AFFINITY
 - > **MEMORY_AFFINTY=MCM**
memory allocation is local per MCM and paging page is global
 - > **MEMORY_AFFINTY=MCM@LRU=EARLY**
both memory allocation is and paging space are local per MCM

No Memory Affinity 0: vmo -o memory_affinity=0 (AIX 5.2)



IBM eServer pSeries IBM

Memory Affinity 1: `vmo -o memory_affinity=1` (AIX 5.2)

MEMORY_AFFINITY not set

↓

**4 kbytes pages is allocated
in round-robin fashion across MCMs**

The diagram illustrates a system with three MCMs (Memory Control Modules) arranged in a triangle, each containing four blue squares representing processors. A central cross indicates interconnectivity. On the left, a vertical bar labeled 'memory' shows a red-hatched segment at the top. On the right, another vertical bar shows a red-hatched segment at the top and another at the bottom. A box at the top contains a red-hatched segment. A box at the bottom contains a red-hatched segment. Arrows point from the top and bottom boxes to the MCMs, and from the left and right bars to the MCMs.

4 kbytes page
(or 16 Mbytes Large Page)

SCICOMP9 - CINECA 2004 © 2004 IBM Corporation

IBM eServer pSeries IBM

Memory Affinity 2: `vmo -o memory_affinity=1` (AIX 5.2)

MEMORY_AFFINITY=MCM

**0) process pages are assigned locally on
MCM pool containing the CPU**

**1) if the process is rescheduled
- memory is not moved**

2) new process pages are assigned locally

The diagram is identical to the one in Memory Affinity 1, showing three MCMs and their connection to system memory. In this configuration, the red-hatched segments in the memory bars and boxes are positioned to represent local allocation to the MCMs.

Default AIX 5.1: `vmtune -y 1`
New in 5.1G: `vmtune -y 2`

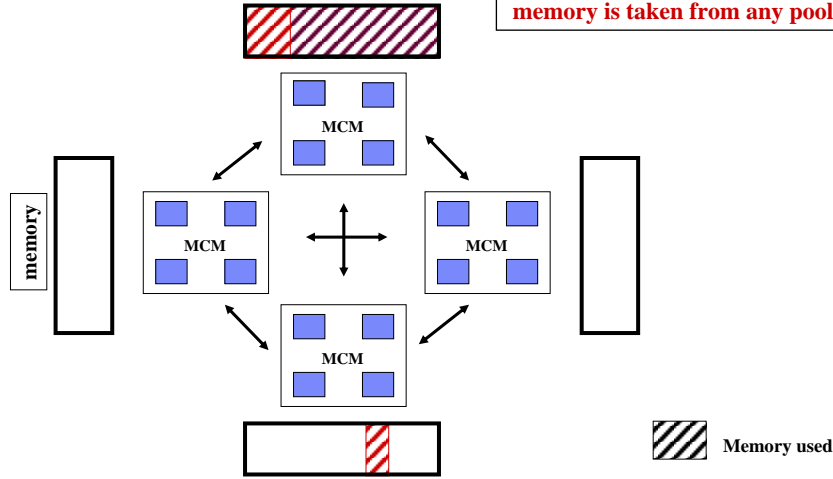
SCICOMP9 - CINECA 2004 © 2004 IBM Corporation

Memory Affinity 3: vmo -o memory_affinity=1 (AIX 5.2)

Not enough memory on the local pool

MEMORY_AFFINITY=MCM

memory is taken from any pool



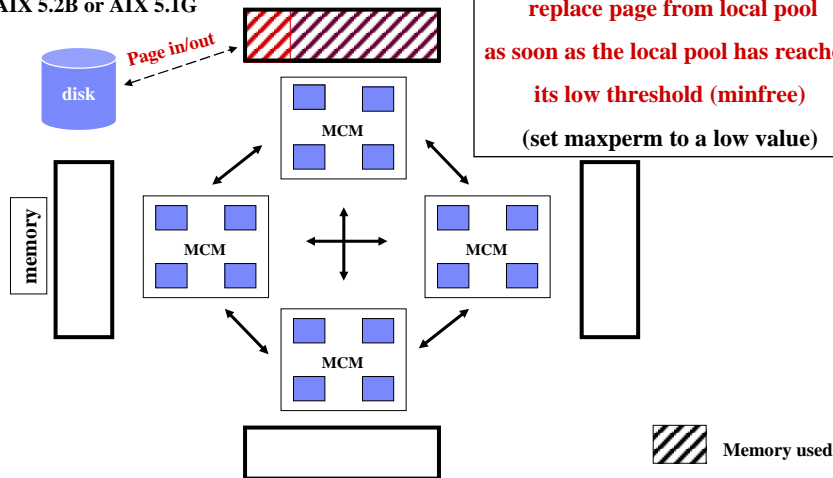
Memory Affinity 4: vmo -o memory_affinity=1 (AIX 5.2, AIX 5.1G)

Not enough memory on the local pool

MEMORY_AFFINITY=MCM@LRU=EARLY

AIX 5.2B or AIX 5.1G

replace page from local pool as soon as the local pool has reached its low threshold (minfree) (set maxperm to a low value)

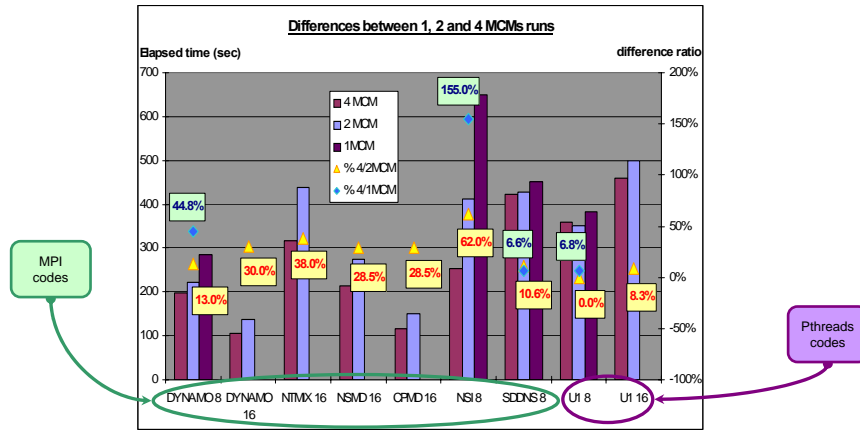


Impact of Memory affinity versus the number of MCM

Elapsed time fluctuations for parallel jobs versus the number of involved MCM

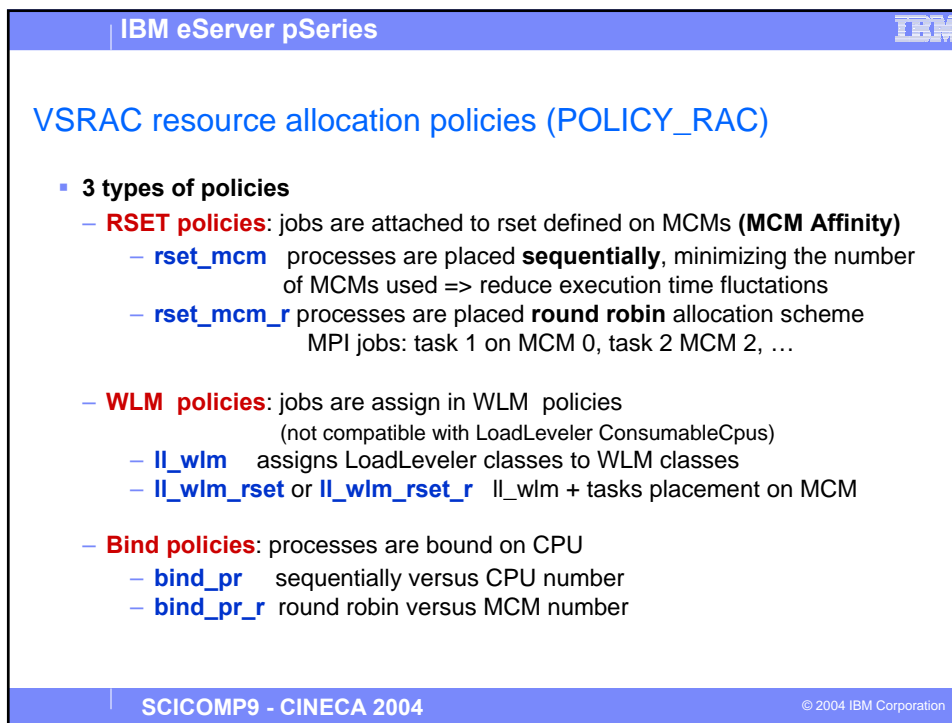
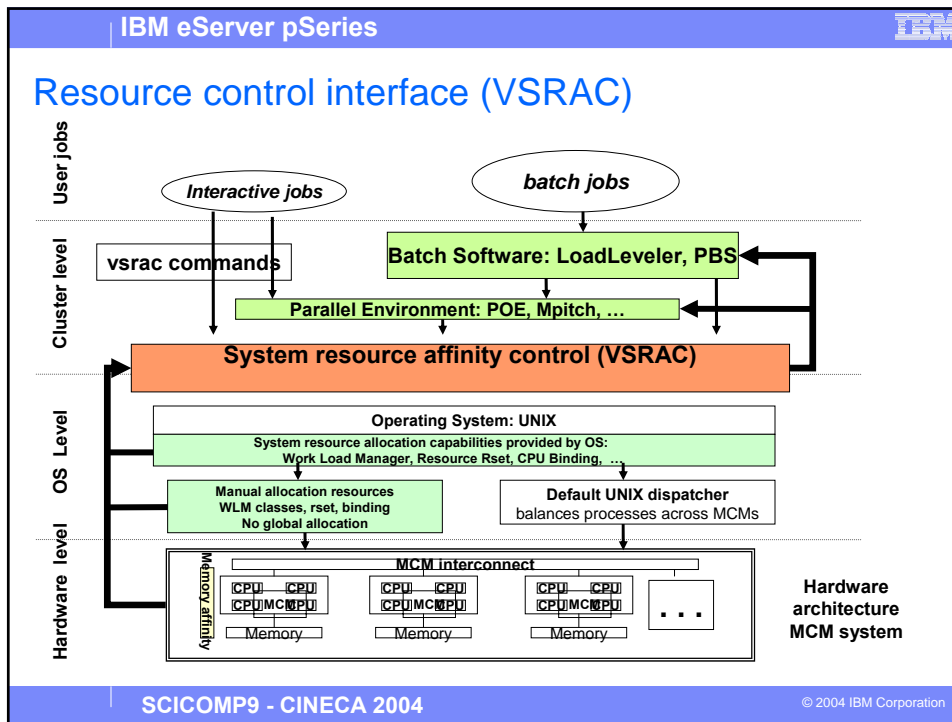
On an idle system, the AIX scheduler spreads processes or threads across MCMs

P690 32 way 1.3 GHz



VSRAC: a tool integrating AIX resource capabilities in HPC environment

- AIX does not provide automatic tool to guarantee MCM Affinity
- In production due to process rescheduling the memory bandwidth can be limited by inter MCM bandwidth (lost of Memory Affinity)
- AIX Affinity services (WLM, binding, rset, Memory affinity) are partially use by HPC environment (Loadleveler, Parallel Environment)
- ...
- **PSSC Solution: VSRAC interface**
 - VSRAC allows multi MCM resource allocation controls in an unique interface including
 - AIX Resources capabilities: WLM, binding, rset, Memory affinity
 - standard resource allocation policies (process placement)
 - Internal workload management
 - Interactive commands
 - Interfaces with LoadLeveler and Parallel environment
 - With environment variables, users or administrator can apply a allocation resource policy
 - Available at <http://tonga.mop.ibm.com/vsrac>



VSRAC Environment variables

- **MCM_AFFINITY [on|off]**: activates VSRAC *tools*
- **POLICY_RAC**: set resource allocation policies
- **JOBTYPE_RAC**: specifies job type (serial, mpi, OpenMP, ...)
- **THREADS_TASK_RAC**: number of threads per process for multi threading job
- **WORKLOAD_RAC [on|off]**: activates VSRAC internal workload management
- **TARGET_RAC**: specifies a list of MCMs or CPUs to limit the job to use only these physical resources
- **MP_PMSUFFIX=vsrac**: ppe variable to set vsrac interface for MPI jobs.

VSRAC usage

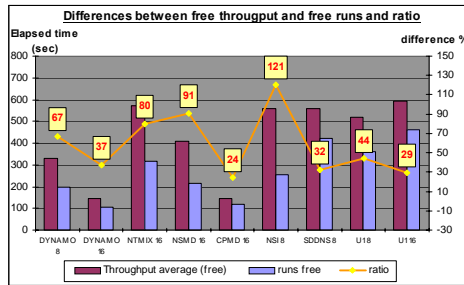
- **Interactive jobs** can be managed by VSRAC under the following conditions:
 - Serial, OpenMP or multithreaded jobs must be started with the **vsrac** driver command
 - > **vsrac** *program_name* *argument1* *argument2* ...
 - MPI jobs must be started with the **mpp** command. mpp is a wrapper script around the poe command that accepts all poe options.
- **LoadLeveler jobs**
 - **Loadleveler configuration**
 - the system administrator must add an user job prolog and epilog in the LoadL_config file.
 - JOB_USER_PROLOG = /opt/vsrac/bin/prolog_vsrac.sh
 - JOB_USER_EPILOG = /opt/vsrac/bin/epilog_vsrac.sh



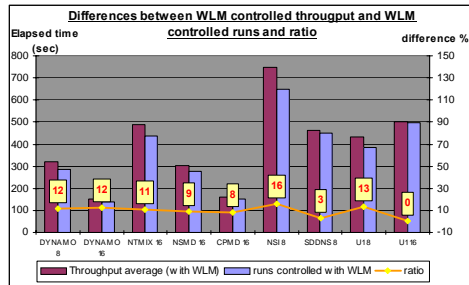
MCM AFFINITY benefits

each application is launched several times with a policy of 1 process/CPU

Without VSRAC control



With VSRAC Control rset_mcm policy



P690 32 way 1.3 GHz